|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学部/院 | 智能与计算学部 | 年级 | 2020级 | 班级 | 网络空间安全1班 |
| 姓名 | 石子跃 | 学号 | 3020244294 | 实验日期 | 2022/5/24 |

实验项目名称 分秒数字钟的设计与实现

**一. 实验目的**

1. 掌握基于 SystemVerilog HDL 的时序逻辑电路建模方法；

2. 掌握计数器设计方法，并能够使用计数器设计使能时钟（用于时钟分频）；

3. 掌握移位寄存器设计方法，并能够利用移位寄存器设计边沿检测电路；

4. 掌握 7 段数码管的动态显示。

**二. 实验内容**

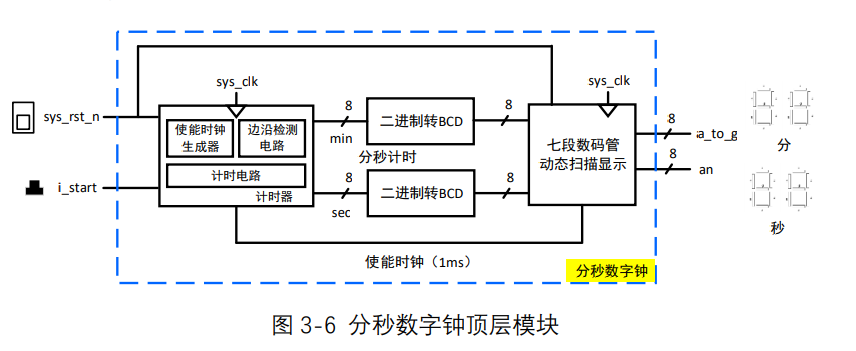
基于 SystemVerilog HDL 设计并实现一个分秒数字钟。整个工程的顶层模块

如图 3-6 所示，输入/输出端口如表 3-1 所示。使用 4 个七段数码管显示当前的

计时。其中，两个数码管对应“分”，另两个数码管对应“秒”。通过 1 个拨动开关

对数字钟进行复位控制。使用 1 个按键对数字中进行“暂停/计时”控制，按键每按

下一次，进行暂停和计时的切换，即暂停时，按下按键启动计时；计时过程中，

按下按键暂停计时。

分秒数字钟由 3 部分构成。

⚫ 第一部分是数字钟的核心——计时器模块，该模块又由3 个子模块构成，

分别是计时电路、使能时钟生成电路和边沿检测电路。

 计时电路通过计数器实现计时功能，产生二进制的“分”（min）和“秒”

（sec）输出。

 使能时钟生成电路用于产生控制七段数码管动态显示的使能时钟，

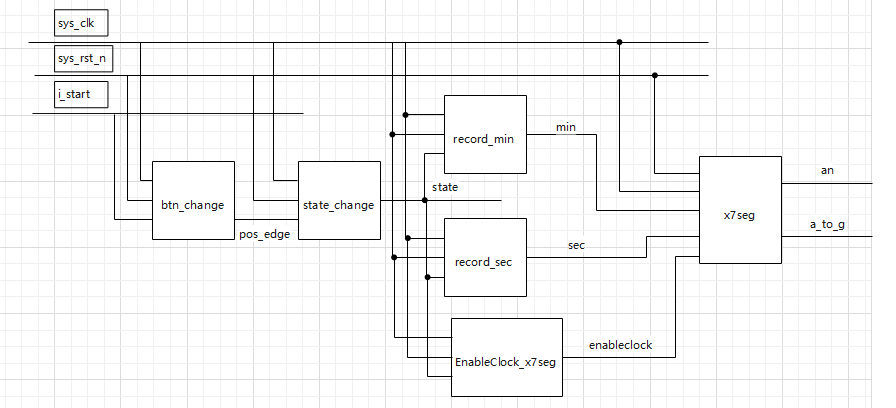
使能时钟高电平出现的周期为 1ms。

 边沿检测电路模块对按键输入进行上升沿检测，产生控制计时器暂

停和启动的信号。

**三．实验原理与步骤（注：步骤不用写工具的操作步骤，而是设计步骤）**

**1. 画出分秒数字钟电路的原理图（模块级别即可，如使能时钟模块、边沿 检测模块等）。**



**2. 分秒数字钟电路中一共使用了几个计数器，作用分别是什么？**

**一共使用了两个计时器，一个用来记录秒数，输出七段数码管最后两位的位置。**

**另一个用来记录分，输出七段数码管前两位的位置。**

**3. 给出分秒数字钟的 SystemVerilog 代码。**

**1. 使能时钟（时钟分频）**

module EnableClock\_x7seg(

input state,

input sys\_clk,

input sys\_rst\_n,

output [1:0]enableclock

);

int counter;

int res;

always\_ff @(posedge sys\_clk)begin

if(!sys\_rst\_n)begin

counter <= 0;

res <= 0;

end

else if(!state)begin

if(counter==24999)begin

counter <= 0;

if(res == 3) res <= 0;

else res <= res +1;

end

else begin

counter <= counter + 1;

end

end

end

assign enableclock = res;

endmodule

按照1ms间隔循环输出，0，1，2，3来控制7段数码管相应位置的显示

**2．边沿检测电路**

module btn\_change(

input sys\_clk,

input i\_start,

output pos\_edge

);

logic dff0\_Q,dff1\_Q;

always\_ff @(posedge sys\_clk)begin

dff0\_Q <= i\_start;

dff1\_Q <= dff0\_Q;

end

assign pos\_edge = dff0\_Q&~dff1\_Q;

endmodule

module state\_change(

input sys\_clk,

input sys\_rst\_n,

input pos\_edge,

output state

);

logic now;

always\_ff @(posedge sys\_clk)begin

if(!sys\_rst\_n) now <= 1;

else begin

if(pos\_edge == 1) now = ~now;

end

end

assign state = now;

endmodule

btn\_change用来检验摁扭变化，state\_change表示一次摁扭变化会导致当前状态的改变

**3.计时器模块**

module record\_sec(

input state,

input sys\_clk,

input sys\_rst\_n,

output [7:0]sec

);

int counter;

logic [7:0]binsec;

always\_ff @(posedge sys\_clk)begin

if(!sys\_rst\_n)begin

counter = 0;

binsec = 0;

end

else if(!state)begin

if(counter==24999999)begin

counter <= 0;

if(binsec == 59) binsec <= 0;

else binsec = binsec + 1;

end

else counter = counter + 1;

end

end

bin2bcd\_0 bcdsec(binsec,sec);

endmodule

1s除以40ns，将其作为计数单位，最后将二进制的秒数转换成8为的bcd码。

**4.总模块**

module dig\_clock(

input sys\_clk,

input sys\_rst\_n,

input i\_start,

output logic [3 : 0] an,

output logic [7 : 0] a\_to\_g

);

logic pos\_edge,state;

logic [1:0]enableclock;

logic [7:0]min,sec;

btn\_change btn(sys\_clk,i\_start,pos\_edge);

state\_change sta(sys\_clk,sys\_rst\_n,pos\_edge,state);

record\_min minute(state,sys\_clk,sys\_rst\_n,min);

record\_sec second(state,sys\_clk,sys\_rst\_n,sec);

EnableClock\_x7seg clock(state,sys\_clk,sys\_rst\_n,enableclock);

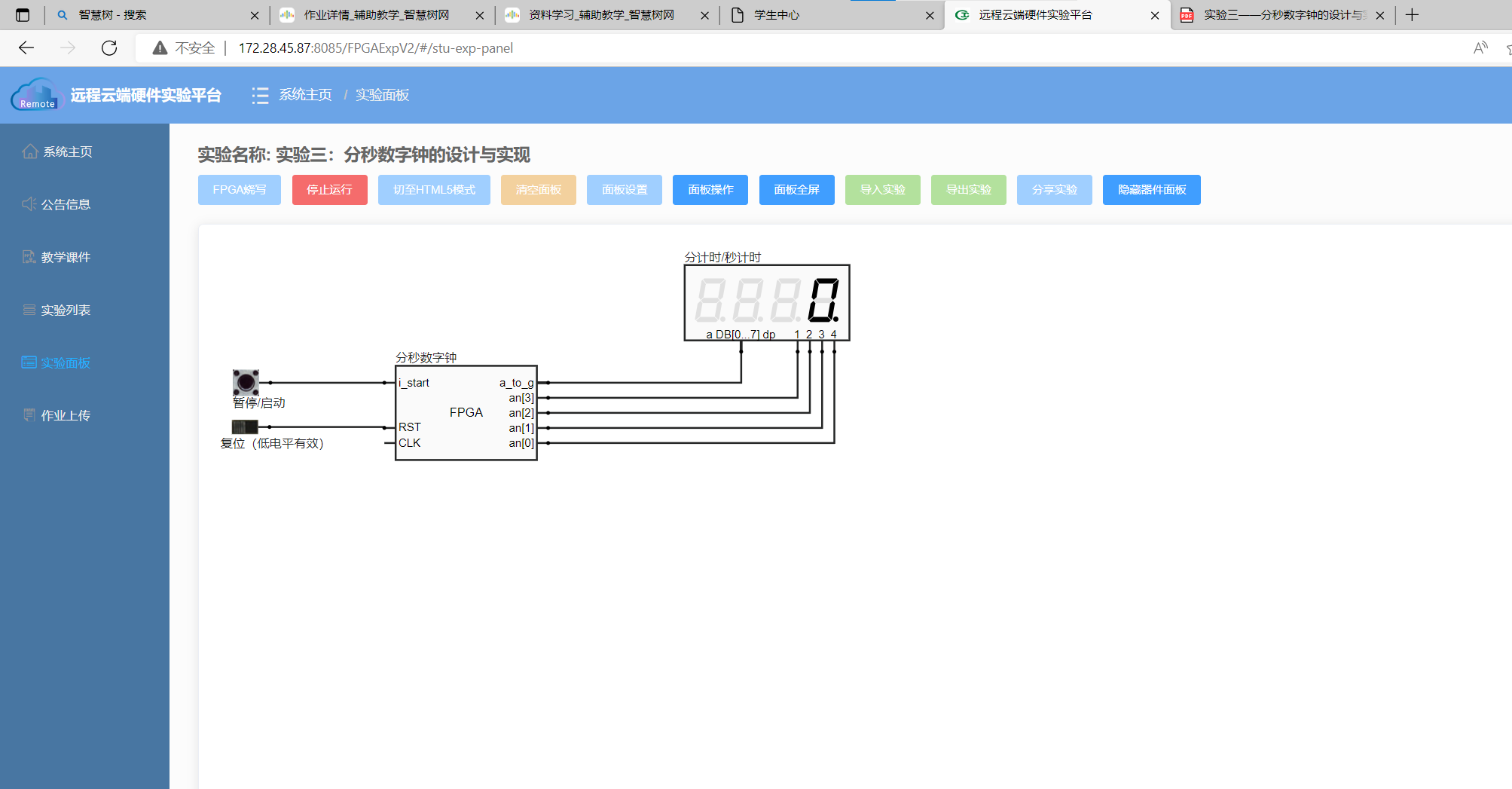
x7seg x7(sys\_clk,sys\_rst\_n,min,sec,enableclock,an,a\_to\_g);

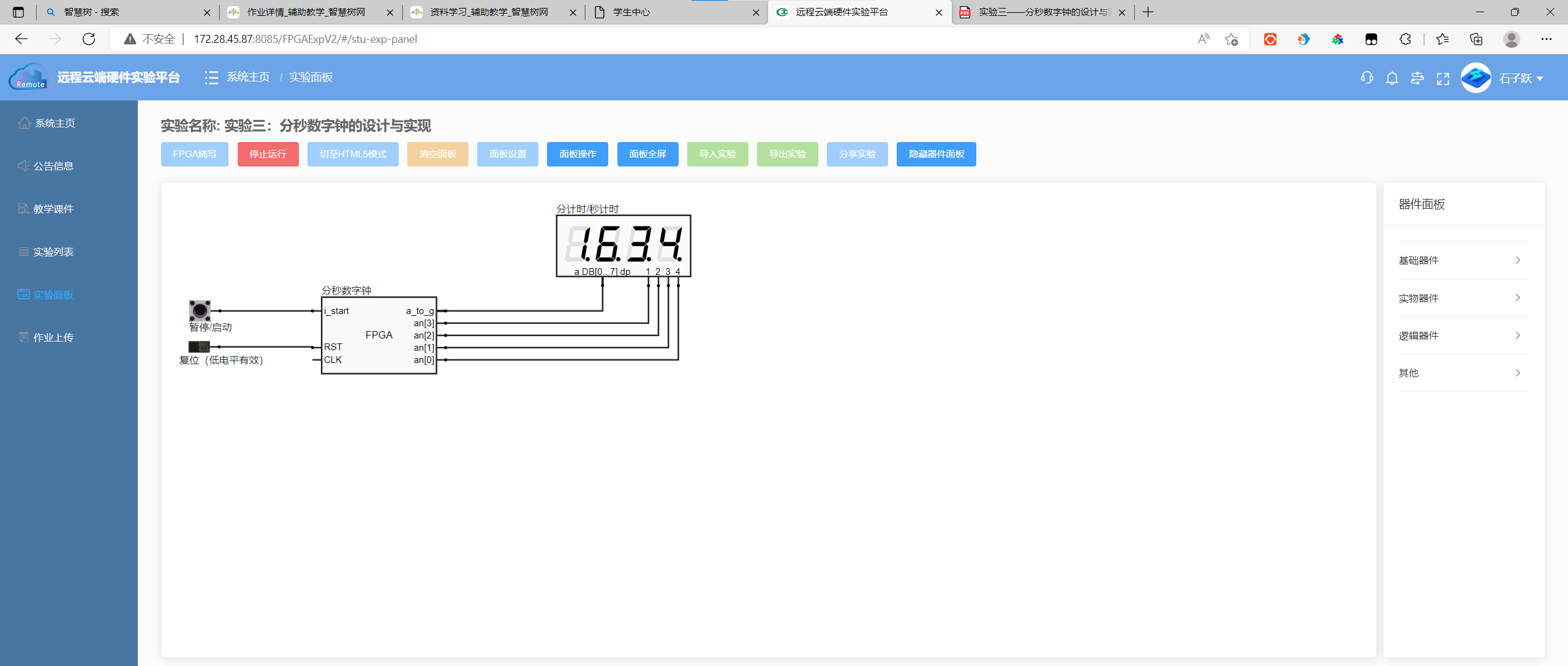
endmodule

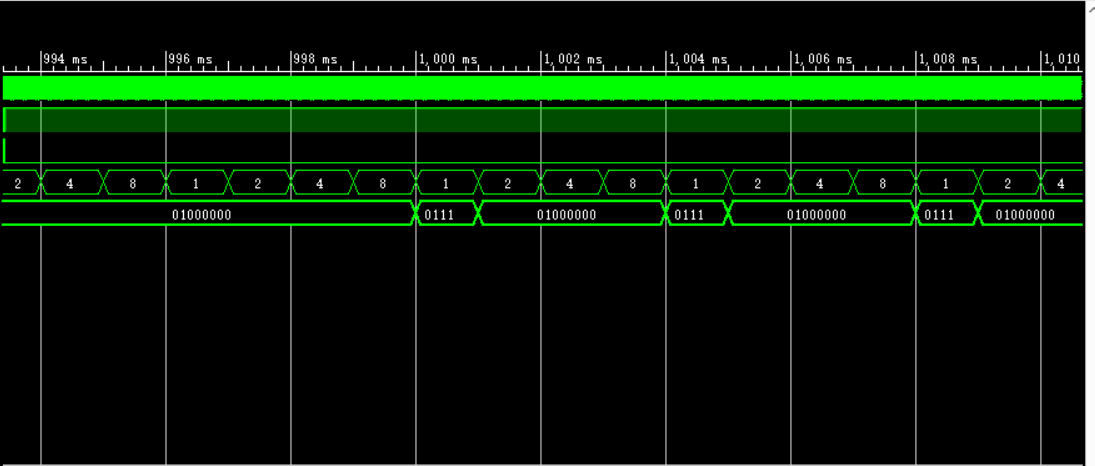
state表示当前计时器状态暂停或开始，enableclock表示当前在七段数码管的哪个位置显示。

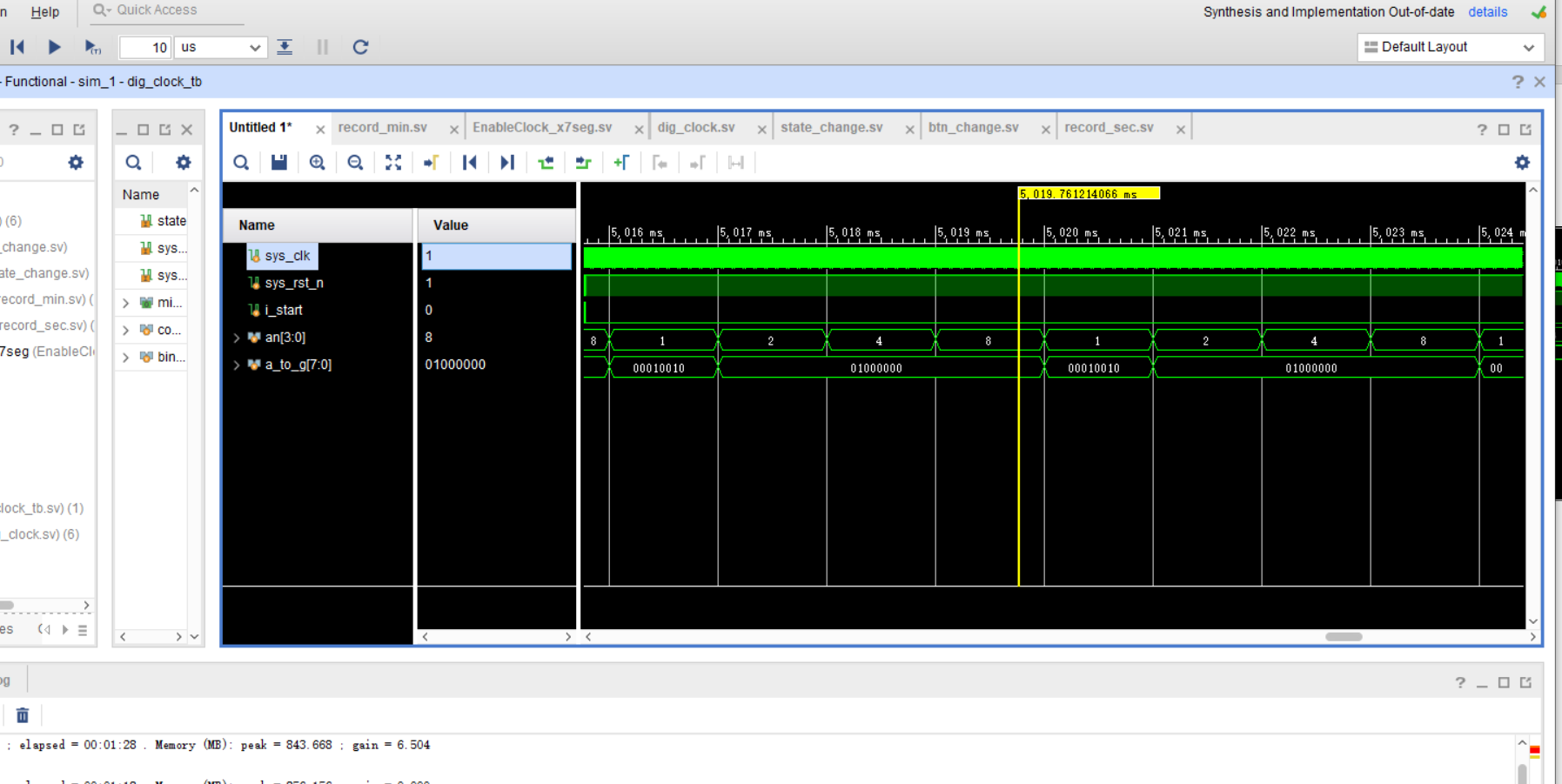
**四．仿真与实验结果（注：仿真需要给出波形图截图，截图要清晰，如果波形过长，可以分段截取；实验结果为远程FPGA硬件云平台的截图）**

注：远程FPGA硬件云平台截图只需要一个测试激励即可









**五．实验中遇到的问题和解决办法**

1.工程框架总体较为繁琐

刚开始就着急进行编程，走了一些弯路。

多阅读实验报告，了解工程结构后，在进行相应编程。

2.行为仿真时速度太慢

对使能时钟频率和计时频率进行适当放大，提高仿真速度

3.暂停时，使能时钟也暂停了

**教师签字：**

**年 月 日**